Defending Against Multiple and Unforeseen Adversarial Videos

Shao-Yuan Lo, *Student Member, IEEE* and Vishal M. Patel, *Senior Member, IEEE*

Abstract—Adversarial robustness of deep neural networks has been actively investigated. However, most existing defense approaches are limited to a specific type of adversarial perturbations. Specifically, they often fail to offer resistance to multiple attack types simultaneously, i.e., they lack multi-perturbation robustness. Furthermore, compared to image recognition problems, the adversarial robustness of video recognition models is relatively unexplored. While several studies have proposed how to generate adversarial videos, only a handful of approaches about the defense strategies have been published in the literature. In this paper, we propose one of the first defense strategies against multiple types of adversarial videos for video recognition. The proposed method, referred to as MultiBN, performs adversarial training on multiple adversarial video types using multiple independent batch normalization (BN) layers with a learning-based BN selection module. With a multiple BN structure, each BN brach is responsible for learning the distribution of a single perturbation type and thus provides more precise distribution estimations. This mechanism benefits dealing with multiple perturbation types. The BN selection module detects the attack type of an input video and sends it to the corresponding BN branch, making MultiBN fully automatic and allow end-to-end training. Compared to present adversarial training approaches, the proposed MultiBN exhibits stronger multi-perturbation robustness against different and even unforeseen adversarial video types, ranging from Lp-bounded attacks and physically realizable attacks. This holds true on different datasets and target models. Moreover, we conduct an extensive analysis to study the properties of the multiple BN structure.

Index Terms—Adversarial video, adversarial robustness, adversarial training, multi-perturbation robustness.

I. INTRODUCTION

RECENT advances in deep learning have led deep neural networks (DNNs) to perform outstandingly well in many computer vision problems [1]–[3], including tasks such as video classification [4]–[6]. However, researchers have shown that DNNs are easily misled when presented by adversarial examples [7], [8]. This is commonly done by adding carefully crafted perturbations to the input data. Various approaches have also been proposed in the literature to defend against adversarial attacks [9]–[16]. Among them, adversarial training [10], [11], [15] is shown to provide stronger robustness especially to the more challenging white-box and adaptive attacks [17]. Therefore, adversarial training has been used as the foundation for more advanced defense techniques. However, present adversarial training approaches usually lead to performance degradation on clean data [18], [19]. Xie et

S.-Y.. Lo and V. M. Patel are with the Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD 21218, USA; e-mail: sylo@jhu.edu, vpatel36@jhu.edu

al. [20] indicated that this problem is due to the distribution mismatch between clean and adversarial examples. In order to deal with this issue, they leveraged an auxiliary batch normalization (BN) layer [21] to disentangle the two distributions. In addition, most existing adversarial training techniques are tailored to one specific perturbation type, e.g., a certain Lp-norm perturbation [11], [22], [23] or physically realizable attacks [24]. A model trained on a specific attack can improve its robustness to that particular attack but often fails to defend when presented with a sample that is perturbed by a different type of attacks [25]. Although there have been several attempts aim to resist multiple attack types, they usually do not consider physically realizable attacks, unforeseen attacks, or how well they perform on clean images [26]–[29]. In a real-world application, the input data could be clean (i.e., unattacked), adversarial, or even attacked with a novel attack that the network has never seen before.

On the other hand, most recent research in this area has focused on static images. Generating adversarial examples and defense methods for videos is relatively less explored. Although a few recent works have extended adversarial attacks to videos [30]–[34], we are aware of few studies so far which delve into detecting or defending against adversarial videos [35], [36]. AdvIT [36] is one of the first adversarial frame detectors based on temporal consistency for videos. However, their approach only detects whether a video has been attacked or not. It does not provide a defense mechanism against the attacked videos. Jia et al. [35] leveraged denoising and frame reconstruction for defense. However, it is not clear how well their defense method works on white-box attacks as it was not reported in [35].

In this paper, we propose MultiBN, which is one of the first defense methods for defending against adversarial videos and considering the accuracy on clean samples as well as the robustness to multiple and unforeseen perturbations. Specifically, we consider four of the most significant types of attacks: projective gradient descent (PGD) [11], rectangular occlusion attack (ROA) [24], adversarial framing (AF) [34] and the proposed salt-and-pepper attack (SPA). Fig. 1 gives an illustration of these attacks on video frames. PGD and ROA are originally designed to attack images. We extend these to videos by perturbing each frame and unveil that video recognition models are also vulnerable to these attacks. SPA is a new video attack we design, which looks like salt-and-pepper noise. PGD and SPA belong to the Lp-bounded attack group while ROA and AF belong to the physically realizable attack group. We select one from each group as the known attack type (PGD and ROA) and leave the others as the unforeseen attack type (AF and SPA), where only the known attacks are used for
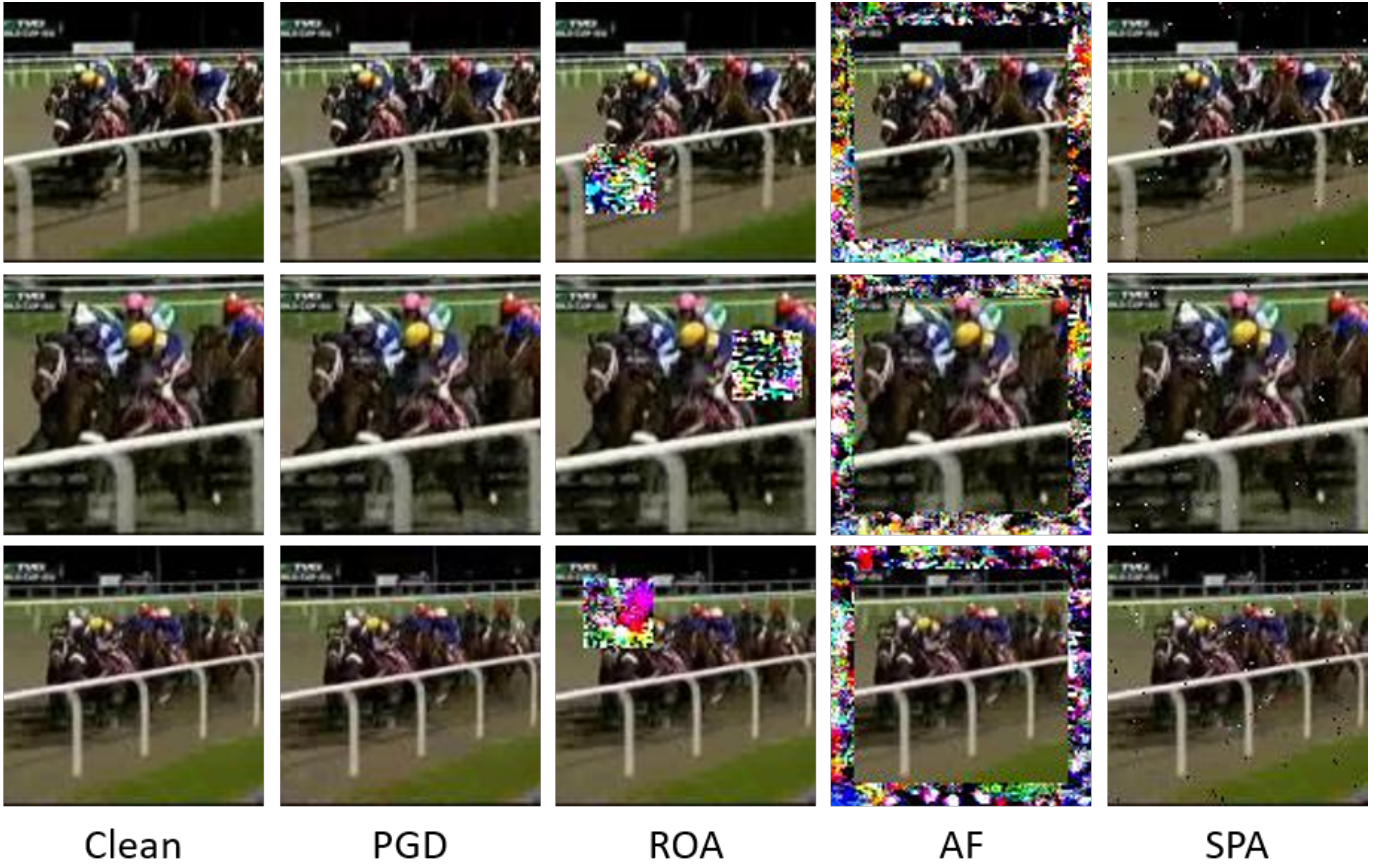
Fig. 1: Illustration of the four types of adversarial videos we consider. Three video frames from the UCF-101 dataset [37] are displayed here.

adversarial training. MultiBN aims to defend against all of these attack types while retaining the performance on clean samples simultaneously.

We first demonstrate that training a model on a specific attack type can gain robustness to that attack and somewhat to another attack in the same group, but typically cannot defend against the attacks in another group. Training models on multiple attack types together (*multi-perturbation training*) improves *multi-perturbation robustness*, yet accuracy on clean samples is sacrificed. This is mainly due to the distribution mismatch among clean and different types of adversarial examples. We assume that the attacks in the same group have a relatively similar distribution. Therefore, inspired by [20], [38], the proposed MultiBN employs multiple BN branches in a single network: for the clean, Lp-norm and physically realizable attack examples, individually. Each BN branches is responsible for learning the distribution of a specific type of examples, which can offer more accurate distribution estimations for these types. Because BN is a lightweight component included in common DNNs, using multiple BN branches causes only minor parameter increases and computational overhead. MultiBN also contains a BN selection module, which detects the attack type of an input video and sends it to the corresponding BN branch, so the entire MultiBN is fully automatic and allows end-to-end training. Compared to existing adversarial training and multi-perturbation training approaches, MultiBN achieves stronger adversarial robustness against multiple, more diverse, and even unforeseen perturbations, while retaining higher accuracy on clean samples. An extensive analysis showing the properties of the multiple BN structure is also presented.

Our main contributions are summarized as follows:

- We propose a novel adversarial defense method, MultiBN, based on a multiple BN structure and a BN selection module. To the best of our knowledge, this is the first defense against multiple and unforeseen adversarial videos.
- The proposed MultiBN achieves stronger multi-perturbation robustness and better clean sample performance than existing multi-perturbation training approaches. This holds true on different datasets and target models.
- We provide an extensive analysis to study the properties of the multiple BN structure under various conditions.

II. RELATED WORK

A. Adversarial Training

Adversarial training is currently considered the most effective defense approach against adversarial perturbations, particularly for the white-box attacks. Goodfellow et al. [7] first

proposed this strategy. They trained DNNs with both clean and adversarial images to improve adversarial robustness. Madry et al. [11] viewed adversarial training from a min-max optimization perspective, training models with solely adversarial images. It has held great promise and has been widely used as a benchmark. Xie et al. [15] developed a feature denoising block, which increases the network capability of handling adversarial training. Xie et al. [36], [38] demonstrated that proper normalization management is important for enhancing robustness and even performance. Zhang et al. [19] introduced TRADES, which uses an alternative loss to perform adversarial training and attained better trade-off between robustness and performance.

Several studies focus on multi-perturbation robustness. Tramèr et al. [26] investigated adversarial robustness to multiple perturbations, including Lp-bounded attacks and rotation-translation attacks. They provided AVG and MAX adversarial training schemes. Maini et al. [27] incorporated multi-perturbation models into a single attack by Multi Steepest Descent (MSD). MSD is robust to different Lp-bounded attacks. Nevertheless, these studies do not take potential unforeseen attack types and clean images into consideration. Laidlaw et al. [29] adversarially trained a target model by Neural Perceptual Threat Model (NPTM), showing good resistance to Lp-bounded attacks and spatial attacks. However, its robustness cannot generalize to physically realizable attacks. Lin et al. [28] aimed to defend against Lp and non-Lp attacks, but they require a pre-constructed On-Manifold dataset, which is too expensive for practical uses. Our MultiBN manages normalization with low costs to enhance the robustness to multiple, more diverse, and even unforeseen perturbations, while retaining higher accuracy on clean images simultaneously.

B. Adversarial Videos

Most existing literature on adversarial attacks and defense are based on static images. There are only a few works that address attacks and defense techniques for videos. Wei et al. [33] are the first to explore adversarial examples in videos. They found that perturbations propagate through video frames in the CNN+RNN based video classifier [5], and thus proposed a temporally sparse attack. Li et al. [31] generated video attacks by a generative model. Zajac et al. [34] developed an attack that keeps frames unchanged and just attaches an adversarial frames on the border of each video frame. Jiang et al. [30] introduced V-BAD for black-box video attacks.

Few studies for detecting or defending against video attacks are presented. Xiao et al. [36] proposed AdvIT based on temporal consistency to detect adversarial frames within a video. However, their approach only detects whether a video has been attacked or not. It does not provide a defense mechanism against the attacked videos. Jia et al. [35] presented a similar detector, along with a temporal defense and a spatial defense. The temporal defense reconstructs perturbed frames with adjacent clean frames. The spatial defense uses and denoises the reconstructed frames to mitigate the effect of adversarial perturbations. However, their approach is only evaluated on the black-box attack setting. It is not clear how

well their defense method works on white-box attacks [17] as it was not reported in [35].

C. Physically Realizable Attacks

Physical attack is a class of adversarial attacks that can be implemented in the physical space. Physically realizable attack refers to the digital representation of physical attacks. Such attacks fool DNNs by modifying physical objects being photographed. Sharif et al. [39] generated printable perturbations inside eyeglass frames to attack face recognition systems. Brown et al. [40] created an adversarial patch that can be put next to a real-world object, making that object be misclassified. Thys et al. [41] further extended the adversarial patch to fooling human detectors. Wu et al. [24] proposed DOA, a defense against physically realizable attacks. DOA performs adversarial training with rectangular occlusion attack (ROA), which places an adversarial rectangular sticker on an image, improving physical robustness. However, it fails to resist Lp-bounded attacks. Our MultiBN is robust to Lp-bounded attacks and physically realizable attacks simultaneously.

III. PRELIMINARIES

A. Multiple Adversarial Video Types

For our investigation, we construct four types of video attacks: L_∞-norm PGD [11], ROA [24], AF [34], and the new SPA attack (see Fig. 1). Among them, L_∞-norm PGD and SPA (L_0-norm) belong to the Lp-bounded attacks; ROA and AF belong to the physically realizable attacks. In our experiments, we set PGD and ROA as the known attack types available for adversarial training, while AF and SPA are used as unforeseen attack types used only during inference. We aim to defend against multiple adversarial video types, including Lp-bounded and physically realizable attacks as well as known and unforeseen attacks. All of these attacks are set to untargeted since the untargeted attack is considered more difficult to resist than the targeted attack.

1) Projective Gradient Descent: PGD attack is defined as:

$$x^{t+1} = \Pi_{x+\mathbb{S}} \left(x^t + \alpha \cdot sign(\bigtriangledown_x L(x, y; \theta)) \right) \qquad (1)$$

where x is a data sample, y is the ground-truth label, θ is model parameters, L is the training loss, and $\mathbb{S}$ denotes the set of allowed perturbations. The perturbation size ϵ is described as $\| x^{t_{max}} - x \|_p \leq \epsilon$, where t_{max} denotes the maximum interation number. PGD is a powerful multi-step variant of the Fast Gradient Sign Method (FGSM) [7]. It has become one of the most important benchmarks in the current adversarial example research [15], [24], [38]. We extend the L_∞-norm PGD from images to videos by taking the gradient descent with respect to an entire input video.

2) Rectangular Occlusion Attack: ROA attack introduces L_∞-norm PGD inside a fixed-size and fixed location rectangle on an image. The size is pre-defined and the location is searched with respect to the highest loss. We extend ROA to videos, in which each frame is perturbed by a rectangle. To save computations, we skip the location search step. Instead, we randomly assign the rectangle location for each video frame and then apply PGD on it.

3) Adversarial Framing: AF attack adds adversarial framings on the border of each video frame while most of the frame pixels are not modified. Specifically, it first fixes the framing location then applies PGD inside it. It is originally designed as a universal attack. To save computations and generate stronger adversaries, we perform a non-universal version, and it can be defined as follows:

$$x^{t+1} = \Pi_{x+\mathbb{S}} \left(x^t + m \cdot \alpha \cdot sign(\bigtriangledown_x L(x,y;\theta)) \right) \quad (2)$$

where $m \in \{0,1\}$ is the AF mask. Let p be a pixel index of **m**. If p is on the border of m within a framing width s_{AF}, $m_p = 1$; otherwise, $m_p = 0$.

4) Salt-and-pepper Attack: Inspired by the one-pixel attack [42], we design a new video attack. For computation saving, instead of using differential evolution, we randomly select a pre-defined number of pixels on each video frame, then apply PGD on those pixels. We consider it as a kind of L_0-norm attack because the number of adversarial pixels is bounded. We name this new attack *salt-and-pepper attack (SPA)*, as the perturbations look like salt-and-pepper noise.

B. Adversarial Training and Multi-perturbation Training

The proposed MultiBN is based on adversarial training and multi-perturbation training. We briefly review adversarial training and state-of-the-art multi-perturbation training schemes to describe the preliminary formulation of our method.

To begin with, we recall the objective function for training a DNN model:

$$\theta^* = \arg \min_\theta \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[L(x,y;\theta) \right], \quad (3)$$

where x is a clean training sample with ground-truth label y in the training set $\mathbb{D}$, θ is model parameters, and L denotes the training loss. Madry's adversarial training [11] applies the min-max optimization and trains models exclusively on adversarial examples:

$$\theta^* = \arg \min_\theta \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\max_{\delta\in\mathbb{S}} L(x+\delta,y;\theta) \right], \quad (4)$$

where δ denotes an adversarial perturbation in the perturbation set $\mathbb{S}$. AdaProp [20] aims to improve the performance on clean samples and trains the model with a mixture of clean data and adversarial examples as follows [7], [10]:

$$\theta^* = \arg \min_\theta \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[L(x,y;\theta) + \max_{\delta\in\mathbb{S}} L(x+\delta,y;\theta) \right]. \quad (5)$$

TRADES [19] uses an alternative objective function for adversarial training:

$$\theta^* = \arg \min_\theta \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[L(x,y;\theta) + \max_{\delta\in\mathbb{S}} L(x+\delta,f(x);\theta) \right], \quad (6)$$

where $f(x)$ is the output vector of the target model with a Softmax operator. In other words, TRADES replaces y with $f(x)$ to compute the cross-entropy loss of adversarial examples.

Regarding multi-perturbation robustness, Tramèr et al. [26] introduced two adversarial training strategies: AVG strategy and MAX strategy. AVG trains on all types of adversarial examples simultaneously and optimizes these adversarial losses together as follows:

$$\theta^* = \arg \min_\theta \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\sum_{i=1}^N \max_{\delta_i\in\mathbb{S}_i} L(x+\delta_i,y;\theta) \right], \quad (7)$$

where N is the number of perturbation types. MAX considers the worst-case attack. It trains on the strongest adversarial example that obtains the maximum loss among all types of attacks:

$$\theta^* = \arg \min_\theta \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[L(x+\delta_k,y;\theta) \right], \quad (8)$$

where

$$\delta_k = \arg \max_{i\in[1,N]} \left[\max_{\delta_i\in\mathbb{S}_i} L(x+\delta_i,y;\theta) \right], \quad (9)$$

which denotes the strongest type of attacks. MSD [27] maximizing the worst-case loss over all the considered perturbations at each projected steepest descent step to construct a single perturbation. This can be described as the follows:

$$\theta^* = \arg \min_\theta \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[L(x+\delta_{MSD},y;\theta) \right], \quad (10)$$

where δ_{MSD} is the constructed single perturbation.

IV. PROPOSED METHOD

In real-world applications, the input data could be clean, adversarial, or even attacked with a novel attack that the network has never seen before. Hence, it is important to design a defense solution that can resist multiple known and unforeseen perturbations while retaining the performance on clean samples. The proposed method, MultiBN, is based on a multiple BN structure and a BN selection module. Fig. 2 gives an overview of MultiBN.

A. Multiple Batch Normalization Structure

Adversarial training on a single perturbation type is generally weakly robust to the other types of attacks. On the other hand, most state-of-the-art DNNs contain BNs [21] in their architecture to normalize input features, which improves performance [3], [6]. However, owing to the different distributions among multiple perturbation types, BNs suffer from the distribution mismatch when multi-perturbation training is conducted, and thus fails to gain promising multi-perturbation robustness. To address this problem, we deploy multiple BN branches into each BN layer of the target model and keep the rest of the parts unchanged, i.e., still a single network [20], [38]. Clean data and each perturbation type used for training are assigned an individual BN branch. Since BN is a lightweight component, multiple BN branches cause only minor parameter increases and computational overhead.

Ideally, each BN branch is responsible for estimating the assigned a single or a family distribution(s), and thus can properly disentangle multiple distributions. Consider adversarial training as a min-max optimization problem [11], for forward
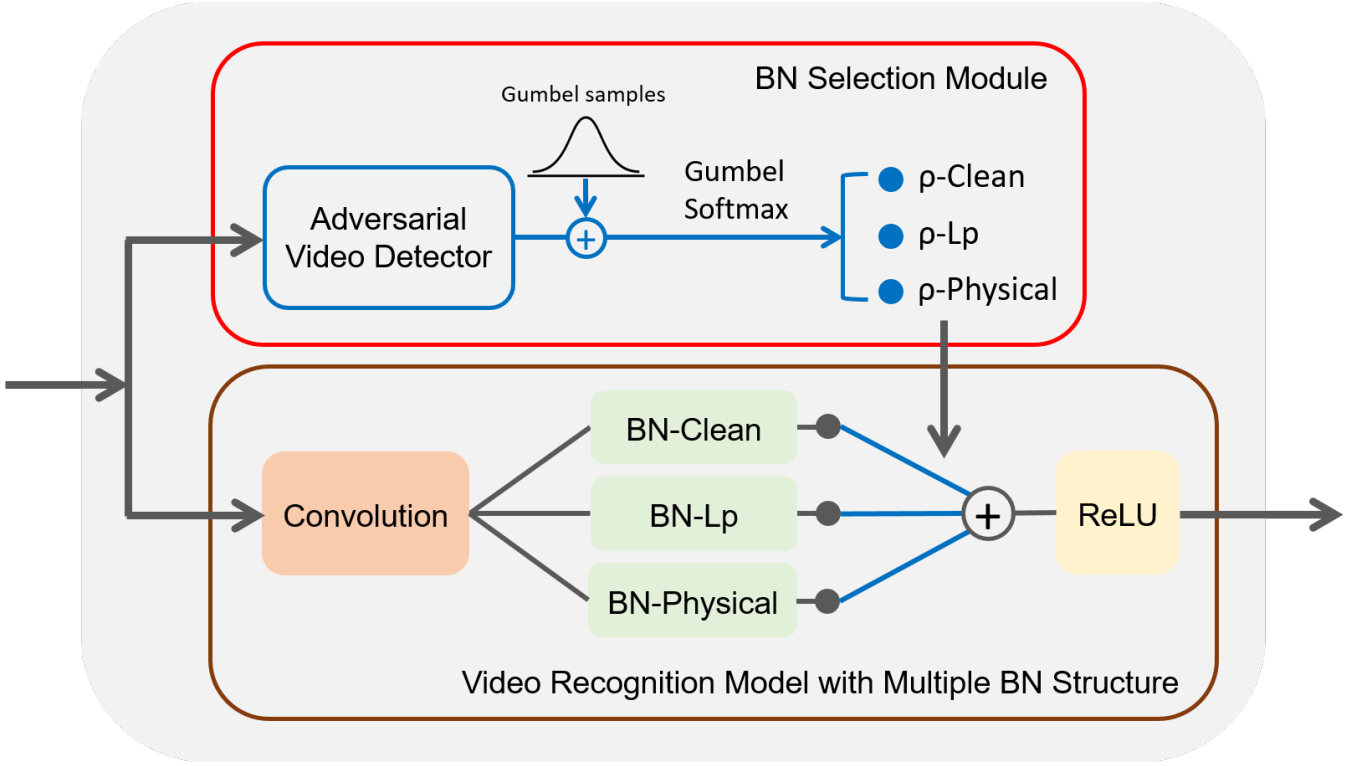
Fig. 2: Overview of MultiBN, the proposed adversarial defense framework. Every batch normalization layer of the video recognition model is replaced by a multiple BN structure. This figure illustrates only one layer for simplicity.

and backward passes, we can manually make each perturbation type attack the target model through its assigned BN branch at the inner maximization step. For the outer minimization step, we send clean inputs or the generated adversarial examples to their corresponding BN branch as well. The ideal objective function can be defined as follows:

$$\theta^* = \arg \min_{\theta} \mathop{\mathbb{E}}_{(x,y) \sim \mathbb{D}}$$
$$\left[L(x, y; \theta^c, \theta^b_0) + \sum_{i=1}^{N} \max_{\delta_i \in \mathbb{S}_i} L(x + \delta_i, y; \theta^c, \theta^b_i) \right], \quad (11)$$

where θ^c is convolution parameters, θ^b_i is the BN parameters of the i-th data type, and $\theta = \theta^c + \sum_{i=0}^{N} \theta^b_i$ denotes all the model parameters.

In practical scenarios, DNNs should provide robustness against unforeseen attacks. An exhaustive investigation is too expensive; instead, we can summarize different attack types into several groups based on their distributions, then build a BN branch for each group. In our case, we deploy three BN branches for clean data, Lp-bounded attacks and physically realizable attacks, respectively (see Fig. 2). Typically, adversarial training on a strong attack has better robustness [11], where PGD and ROA are good representatives of Lp-bounded attacks and physically realizable attacks, respectively. Therefore, we train a target model on clean, PGD, and ROA examples using the 3-BN structure with Eq. (11), where $N = 2$.

B. Batch Normalization Selection Module

At inference time, we cannot control the data flow, so the input data have to pass through the corresponding BN branch automatically. To this end, we propose a BN selection module based on an adversarial video detector and a Gumbel-Softmax operator [43], [44] (see Fig. 2). The adversarial video detector is achieved by a $(N + 1)$-class video classification model, where N is the number of attack types used for training. It is trained to not only identify whether an input video is clean or adversarial but also recognize the attack types. In our case, the detector is trained with $N = 2$ on clean, PGD and ROA examples to recognize clean data, Lp-bounded attacks and physically realizable attacks. Hence, unforeseen perturbations would also be classified into the most similar attack group.

Intuitively, we can build a switch module to send the input to the proper BN branch according to its detection result. However, the $argmax$ operation, which applies to the adversarial video detector's logits for BN branch selection, is not differentiable. This makes end-to-end training infeasible. To address this issue, we leverage the Gumbel-Softmax trick to allow the gradients to backpropagate through a discrete sampling process [45]. Specifically, we approximate $argmax$ by the differentiable and continuous Gumbel-Softmax function, which is defined as follows:

$$\rho_i = \frac{exp((\log \pi_i + G_i) / \tau)}{\sum_{j=1}^{k} exp((\log \pi_j + G_j) / \tau)}, \quad (12)$$

where $\pi_1, ..., \pi_k$ are the adversarial video detector's logits, $G_1, ..., G_k$ are i.i.d. Gumbel samples, τ is the Softmax tem-

TABLE I: Results (%) of MultiBN-manual on target model 3D ResNeXt-101 and dataset UCF-101. No Defense is trained on only clean data. AT-PGD, AT-ROA, AT-AF and AT-SPA are adversarially trained on a single specific attack type. The best results are in bold, and the best results among adversarially trained models are underlined.

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**89.0**	3.3	0.5	1.6	8.4	20.6	0.0
AT-PGD	78.6	**49.0**	5.0	0.6	67.1	40.1	0.3
AT-ROA	82.6	12.5	**69.0**	54.0	17.6	47.1	7.9
AT-AF	84.6	7.1	3.9	**80.5**	12.2	37.7	2.1
AT-SPA	83.5	36.9	2.6	0.7	**69.5**	38.6	0.2
MultiBN-manual	<u>83.7</u>	<u>46.4</u>	<u>65.6</u>	<u>57.0</u>	<u>60.4</u>	**62.6**	**40.7**

perature, and $k = 3$ in our case. Next, $\rho_1, ..., \rho_k$ are used as ratio factors to weight each BN branch's output features:

$$\tilde{h} = \sum_{i=1}^{k} \rho_i \, h_i, \tag{13}$$

where $h_1, ..., h_k$ are each BN branch's output features, and $\tilde{h}$ is the weighted feature that would be the input of the next network component. In this way, the correct BN branch's output feature h_* would dominate $\tilde{h}$, making $\tilde{h}$ a good approximation of h_*.

C. The Entire Framework

With the BN selection module, MultiBN can operate automatically during inference without manual control, and it enables end-to-end training since the entire framework is differentiable. The end-to-end training objective is as follows:

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[L(x, y; \theta) + \lambda \cdot L(x, y^{det}; \theta^{det}) \right.$$
$$\left. + \sum_{i=1}^{N} \left(\max_{\delta_i \in \mathbb{S}_i} L(x + \delta_i, y; \theta) + \lambda \cdot L(x + \delta_i, y^{det}; \theta^{det}) \right) \right], \tag{14}$$

where θ is the entire MultiBN's parameters, $\theta^{det} \in \theta$ is the adversarial video detector's parameters, y^{det} is the ground-truth of the video types for training the adversarial video detector, and λ is a trade-off hyperparameter. Using Eq. (14) jointly optimizes the target model with the multiple BN structure and the BN selection module.

V. EXPERIMENTS

In this section, we first describe our experimental setup. Second, we evaluate MultiBN's manually-controlled version to validate the effectiveness of the multiple BN structure and explore the properties of this structure. Next, we test the proposed MultiBN's robustness and performance and compare it with state-of-the-art multi-perturbation training approaches. Finally, we conduct further analysis on MultiBN's robustness against adaptive attacks and different attack budgets.

TABLE II: Results (%) of each BN branch on the five input types. BN-Clean, BN-Lp and BN-Physical are the clean, PGD and ROA BN branches in the multiple BN structure, respectively.

BN Branch	Clean	PGD	ROA	AF	SPA
BN-Clean	**83.7**	21.3	13.5	5.9	23.8
BN-Lp	79.0	**46.4**	7.7	1.9	**60.4**
BN-Physical	83.0	23.5	**65.6**	**57.0**	26.6

A. Experimental Setup

1) Datasets: We use UCF-101 [37] and HMDB-51 [46] for evaluation, which are widely used video datasets in action recognition. UCF-101 consists of 13,320 videos from 101 action classes, and HMDB-51 has 6,766 videos from 51 action classes. Following [33], we resize their frame dimensinos to 112×112 and uniformly sample each video into 40 frames. UCF-101 is the default dataset if not otherwise specified.

2) Attack Setting: We consider L_∞-norm PGD [11], ROA [24], AF [34], and the proposed SPA attack. For PGD, we set the perturbation size ϵ to $4/255$; for ROA, we set the rectangle size s_{ROA} to 30×30 and ϵ to $255/255$; for AF, we set the framing width s_{AF} to 10 and ϵ to $255/255$; for SPA, we set the number of adversarial pixels on each frame s_{SPA} to 100 and ϵ to $255/255$. The number of attack iterations t_{max} is set to 5 for all the four attacks. To test the proposed method by strong attacks, all of these attacks are untargeted attacks and in the white-box setting (i.e., the attacker has full knowledge of the target model, including the multiple BN structure and the BN selection module).

3) Implementation Details: We choose 3D ResNext-101 and 3D Wide ResNet-50 [6] as our target models, as they are two of the most top-performing 3D CNNs for video recognition, where 3D ResNext-101 is the default target model if not otherwise specified. For our adversarial video detector, we choose the lightweight 3D ResNet-50. We use the pre-trained weights from [6] and conduct adversarial training upon the pre-trained models. We set MultiBN's Softmax temperature $\tau = 1$ and the trade-off hyperparameter $\lambda = 0.1$. All the models are trained by a SGD optimizer with initial learning rate $5e^{-4}$, momentum 0.9 and weight decay $1e^{-5}$, where the learning rate is decreased by a factor of 10 in the middle of the training process.

TABLE III: Results (%) of the cases that the target BN and the inference BN are different.

Attack Inference BN \ Target BN	PGD			ROA		
	BN-Clean	BN-Lp	BN-Physical	BN-Clean	BN-Lp	BN-Physical
BN-Clean	21.3	50.9	35.9	13.5	17.9	56.6
BN-Lp	**72.6**	46.4	**70.5**	30.4	7.7	48.2
BN-Physical	46.4	**52.3**	23.5	**78.5**	**76.4**	**65.6**

Attack Inference BN \ Target BN	AF			SPA		
	BN-Clean	BN-Lp	BN-Physical	BN-Clean	BN-Lp	BN-Physical
BN-Clean	5.9	7.3	49.0	23.8	55.7	41.8
BN-Lp	16.4	1.9	33.5	**77.2**	**60.4**	**75.8**
BN-Physical	**75.2**	**62.5**	**57.0**	49.8	57.4	26.6

We apply the mean accuracy and the union accuracy as the metrics to evaluate the multi-perturbation robustness. The union accuracy requires that the target models need to correctly classify an input sample under all the considered input types.

B. Multiple BN Structure

We first manually select the correct BN branches to investigate the effectiveness of the multiple BN structure. We call this variant MultiBN-manual. Then, we compare MultiBN-manual with vanilla adversarial training [11] that trains on a single attack type. Table I shows that models trained on a specific attack always have the best robustness to that attack. AT-PGD and AT-ROA also yield high robustness to another attack in their own group, showing better generalization. However, all of them almost fail to defend against the attacks from other groups.

MultiBN-manual uniformly achieves the second-highest accuracy across all the five input types, and most of these accuracies are close to the best one. Although MultiBN-manual is not the best from the perspective of any specific input type, it sustains a much better balance when multiple input types are considered. This shows the effectiveness of the multiple BN structure in multi-perturbation robustness. As can be seen, MultiBN-manual's mean accuracy and union accuracy are significantly higher than adversarial training on a single attack.

C. Analysis of Differnt BN Branches

In the previous subsection, the attacker generates perturbations through the BN branch corresponding to its type. During inference, the input is also sent to that BN branch accordingly. In this subsection, we investigate the cases that perturbations are produced on a BN branch which is different from the group of the used attack type, and the input is sent to that BN branch during inference.

In Table II, we can see that each BN branch performs the best on the input type which they are trained on. Moreover, for the unforeseen attacks, BN-Lp is the most robust to SPA, and BN-Physical is the most robust to AF. This result is consistent to our assumption that feeding an unforeseen adversarial example to the BN branch of the same or the most similar group can enjoy the best benefit. Our grouping follows the observation that PGD and SPA have similar distributions (Lp-bounded attacks), and ROA and AF have similar distributions (physically realizable attacks).

D. Analysis of Target BN and Inference BN

We further delve into the cases that an adversarial example is made inference on a BN branch (inference BN) different from the BN branch that is used to generate the adversarial example (target BN). In other words, we consider the cases that the target BN and the inference BN are different.

The results in Table III are mostly consistent with that in Table II, in which BN-Lp has the strongest robustness to Lp-bounded attacks, and BN-Physical has the strongest robustness to physically realizable attacks. PGD attack is an exception: When the target BN is BN-Lp, inference BN-Lp performs the worst.

In addition, we observe that for any specific inference BN, it is more robust to the adversarial examples generated on another BN branch, i.e., target BN and inference BN are different. In such a case, the attack is not a rigorous white-box attack, so we treat it as a kind of gray-box attack, in which the attacker does not know which BN branch would the adversarial example pass through during inference. This unveils that the attacks cannot perfectly transfer to other BN branches though the rest of the model parameters are shared in the same network. Such results show that the multiple BN structure can make secure against this gray-box setting.

E. The Entire MultiBN Framework

In this subsection, we evaluate the entire MultiBN and compare it with state-of-the-art adversarial training and multi-perturbation training approaches, including TRADES [19], AVG [26], MAX [26] and MSD [27]. For TRADES, we apply the AVG strategy to it for multi-perturbation training. Because we take clean data accuracy into consideration, we adjust AVG, MAX and MSD by involving clean data in training. That is, we add the clean data loss term $L(x, y; \theta)$ into the expectation of objective functions Eq. (7), Eq. (8) and Eq. (10).

Table IV reports the results on target model 3D ResNext-101 and dataset UCF-101, Table V reports the results on target model 3D Wide ResNet-50 and dataset UCF-101, Table VI reports the results on target model 3D ResNext-101 and dataset HMDB-51. As expected, No Defense still achieves

TABLE IV: Results (%) of MultiBN and state-of-the-art approaches on target model 3D ResNeXt-101 and dataset UCF-101. The best results are in bold.

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**89.0**	3.3	0.5	1.6	8.4	20.6	0.0
TRADE [19] (ICML'19)	82.3	29.0	5.7	3.3	42.2	32.5	1.9
AVG [26] (NeurIPS'19)	68.9	38.1	51.4	18.5	49.6	45.3	17.3
MAX [26] (NeurIPS'19)	72.8	32.5	31.0	5.8	49.4	38.3	5.5
MSD [27] (ICML'20)	70.2	43.2	1.7	1.6	**56.0**	34.6	0.7
MultiBN (ours)	74.2	**44.6**	**58.6**	**44.3**	53.7	**55.1**	**34.8**

TABLE V: Results (%) of MultiBN and state-of-the-art approaches on target model 3D Wide ResNet-50 and dataset UCF-101. The best results are in bold.

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**88.4**	11.5	0.2	1.0	10.0	22.2	0.0
TRADE [19] (ICML'19)	81.1	26.7	1.1	0.7	39.2	29.8	0.1
AVG [26] (NeurIPS'19)	74.5	43.1	55.6	3.5	57.2	46.8	3.5
MAX [26] (NeurIPS'19)	76.0	32.5	12.2	2.3	39.2	32.4	1.9
MSD [27] (ICML'20)	71.0	46.3	2.9	0.9	**61.1**	36.4	0.2
MultiBN (ours)	77.4	**46.5**	**59.9**	**48.1**	56.7	**57.7**	**37.8**

TABLE VI: Results (%) of MultiBN and state-of-the-art approaches on target model 3D ResNeXt-101 and dataset HMDB-51. The best results are in bold.

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**65.1**	0.0	0.0	0.0	0.3	13.1	0.0
TRADE [19] (ICML'19)	54.8	6.8	0.3	0.0	20.5	16.5	0.0
AVG [26] (NeurIPS'19)	39.0	14.3	17.1	2.8	26.2	19.9	1.4
MAX [26] (NeurIPS'19)	48.6	13.9	16.0	0.1	30.3	21.8	0.0
MSD [27] (ICML'20)	41.4	18.2	0.1	0.0	**31.2**	18.2	0.0
MultiBN (ours)	51.1	**22.0**	**23.7**	**7.8**	29.9	**26.9**	**5.0**

the best performance on clean data, showing that adversarial training degrades clean data performance. TRADES has the best clean data performance among the adversarial training approaches, but it lacks multi-perturbation robustness. AVG improves multi-perturbation robustness to a large extent, yet its clean data performance is very low. MAX is less robust than AVG in our case. MSD has the best and the second-best robustness against SPA and PGD, respectively, but it is vulnerable to physically realizable attacks. The proposed MultiBN achieves the second-best clean data performance among the defenses, the second-best robustness against SPA, and the best robustness against the rest of the attack types. MultiBN consistently outperforms the competitors in terms of mean accuracy and union accuracy by a wide margin, showing great multi-perturbation robustness. This holds true on different datasets and target models.

F. Adaptive Attacks

To thoroughly evaluate MultiBN, we construct an adaptive attack [47], which jointly attacks the target model part and the BN selection module part. The intuition is to generate adversarial examples which can also fool the BN selection
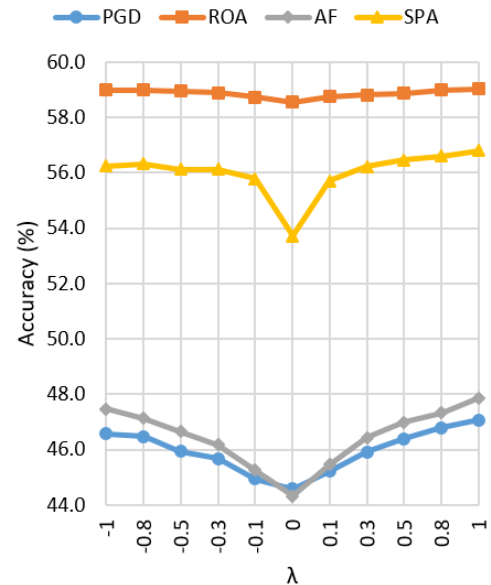

Fig. 3: Results (%) of MultiBN under the adaptive attacks with varied λ of the four attack types.
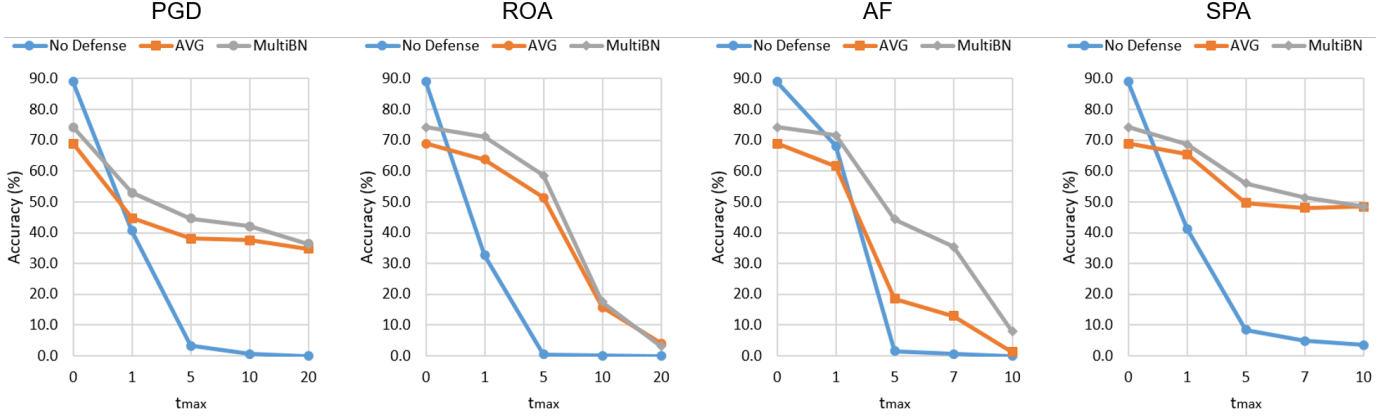
Fig. 4: Results (%) under the four attack types with varied numbers of attack iterations t_{max}.
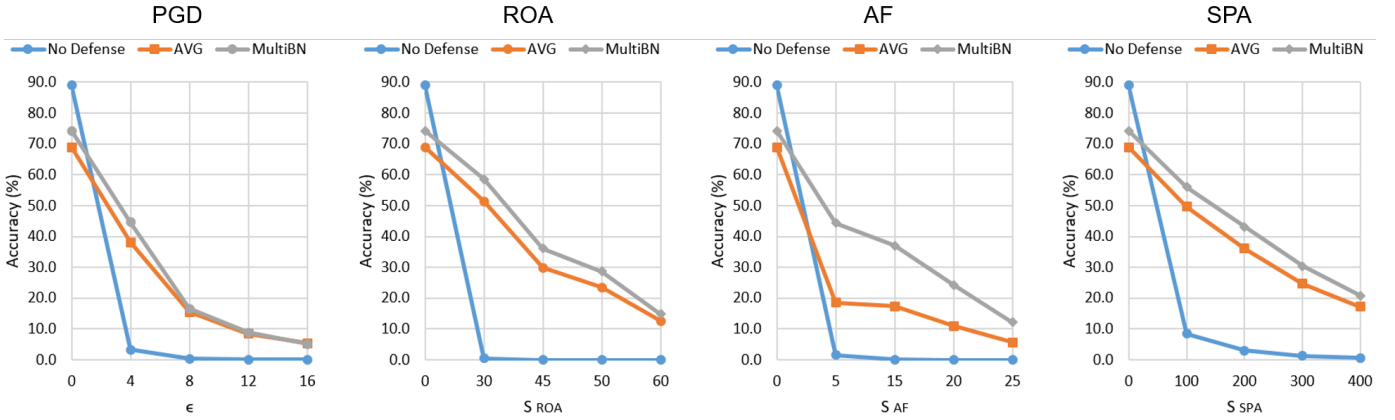

Fig. 5: Results (%) under the four attack types with varied perturbation bounds.

module to let it select the incorrect BN branch, and thus become easier to fool the target model. This adaptive attack is formulated as follows:

$$\delta = \arg\max_{\delta \in \mathbb{S}} \left[L(x + \delta, y; \theta) + \lambda \cdot L(x + \delta, y^{det}; \theta^{det}) \right]. \quad (15)$$

As presented in Fig. 3, the canonical attack has the greatest attacking strength. The accuracies under all the four attack types monotonously increase as $|\lambda|$ increases. This shows that the considered adaptive attack fails to break MultiBN.

G. Attack Budgets

To further evaluate the effectiveness of MultiBN, we test its scalability to different attack budgets. We vary the attack budgets by two aspects: The number of attack iterations t_{max} and the perturbation bounds of different attack types, i.e., PGD's ϵ, ROA's s_{ROA}, AF's s_{AF} and SPA's s_{SPA}. The results are presented in Fig. 4 and Fig. 5. The No Defense baseline and the strongest competitor AVG are compared.

We can see that MultiBN consistently achieves better robustness against different attack types with various attack iterations and perturbation bounds. This shows that MultiBN's multi-perturbation robustness is scalable to various attack budgets.

VI. Conclusions

In this paper, we proposed MultiBN, a new adversarial defense method aiming at multi-perturbation robustness. This is one of the first defenses against multiple and unforeseen adversarial videos. MultiBN uses a multiple BN structure to solve the distribution mismatch problem during multi-perturbation training. A BN selection module makes the entire framework automatic at inference time and differentiable for end-to-end training. Compared to existing adversarial training approaches, MultiBN achieves stronger multi-perturbation robustness against different and even unforeseen Lp-bounded attacks and physically realizable attacks. This holds true on different datasets and target models. Furthermore, we conduct an extensive analysis to explore the properties of the multiple BN structure under various conditions.

Acknowledgment

References

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," in *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[2] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *IEEE International Conference on Computer Vision*, 2017.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[4] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[5] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[6] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[8] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[9] X. Jia, X. Wei, X. Cao, and H. Foroosh, "Comdefend: An efficient image compression model to defend adversarial examples," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[10] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *International Conference on Learning Representations*, 2017.

[11] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[12] E. Raff, J. Sylvester, S. Forsyth, and M. McLean, "Barrage of random transforms for adversarially robust defense," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[13] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," in *International Conference on Learning Representations*, 2018.

[14] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," in *International Conference on Learning Representations*, 2018.

[15] C. Xie, Y. Wu, L. van der Maaten, A. Yuille, and K. He, "Feature denoising for improving adversarial robustness," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[16] S. Addepalli, A. Baburaj, G. Sriramanan, and R. V. Babu, "Towards achieving adversarial robustness by enforcing feature consistency across bit planes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[17] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International Conference on Machine Learning*, 2018.

[18] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," in *International Conference on Learning Representations*, 2019.

[19] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International Conference on Machine Learning*, 2019.

[20] C. Xie, M. Tan, B. Gong, J. Wang, A. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015.

[22] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," in *International Conference on Learning Representations*, 2018.

[23] E. Wong and J. Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International Conference on Machine Learning*, 2018.

[24] T. Wu, L. Tong, and Y. Vorobeychik, "Defending against physically realizable attacks on image classification," in *International Conference on Learning Representations*, 2020.

[25] Y. Sharma and P.-Y. Chen, "Attacking the madry defense model with l_1-based adversarial examples," in *International Conference on Learning Representations Workshop*, 2018.

[26] F. Tramèr and D. Boneh, "Adversarial training and robustness for multiple perturbations," in *Conference on Neural Information Processing Systems*, 2019.

[27] P. Maini, E. Wong, and J. Z. Kolter, "Adversarial robustness against the union of multiple perturbation models," in *International Conference on Machine Learning*, 2020.

[28] W.-A. Lin, C. P. Lau, A. Levine, R. Chellappa, and S. Feizi, "Dual manifold adversarial robustness: Defense against lp and non-lp adversarial attacks," in *Conference on Neural Information Processing Systems*, 2020.

[29] C. Laidlaw, S. Singla, and S. Feizi, "Perceptual adversarial robustness: Defense against unseen threat models," in *International Conference on Learning Representations*, 2021.

[30] L. Jiang, X. Ma, S. Chen, J. Bailey, and Y.-G. Jiang, "Black-box adversarial attacks on video recognition models," in *ACM International Conference on Multimedia*, 2019.

[31] S. Li, A. Neupane, S. Paul, C. Song, S. V. Krishnamurthy, A. K. Roy-Chowdhury, and A. Swami, "Stealthy adversarial perturbations against real-time video classification systems." in *Network and Distributed System Security Symposium*, 2019.

[32] X. Wei, S. Liang, N. Chen, and X. Cao, "Transferable adversarial attacks for image and video object detection," in *International Joint Conferences on Artificial Intelligence*, 2019.

[33] X. Wei, J. Zhu, S. Yuan, and H. Su, "Sparse adversarial perturbations for videos," in *AAAI Conference on Artificial Intelligence*, 2019.

[34] M. Zajac, K. Zołna, N. Rostamzadeh, and P. O. Pinheiro, "Adversarial framing for image and video classification," in *AAAI Conference on Artificial Intelligence*, 2019.

[35] X. Jia, X. Wei, and X. Cao, "Identifying and resisting adversarial videos using temporal consistency," *arXiv preprint arXiv:1909.04837*, 2019.

[36] C. Xiao, R. Deng, B. Li, T. Lee, B. Edwards, J. Yi, D. X. Song, M. Liu, and I. Molloy, "Advit: Adversarial frames identifier based on temporal consistency in videos," in *IEEE International Conference on Computer Vision*, 2019.

[37] K. Soomro, A. R. Zamir, M. Shah, K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[38] C. Xie and A. Yuille, "Intriguing properties of adversarial training at scale," in *International Conference on Learning Representations*, 2020.

[39] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *ACM Conference on Computer and Communications Security*, 2016.

[40] T. Brown, D. Mane, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," in *Conference and Workshop on Neural Information Processing Systems Workshop*, 2017.

[41] S. Thys, W. Van Ranst, and T. Goedemé, "Fooling automated surveillance cameras: adversarial patches to attack person detection," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2019.

[42] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," in *IEEE Transactions on Evolutionary Computation*, 2019.

[43] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *International Conference on Learning Representations*, 2017.

[44] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," in *International Conference on Learning Representations*, 2017.

[45] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, and F. Ren, "Learning in the frequency domain," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[46] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," in *IEEE International Conference on Computer Vision*, 2011.

[47] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," in *Conference on Neural Information Processing Systems*, 2020.